```
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFFFFFFFFFFF      000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFFFFFFFFFFF      000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFFFFFFFFFFF      000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFF               000         000 RRR    RRR      RRR    RRR            TTT        LLL
FFF               000         000 RRR    RRR      RRR    RRR            TTT        LLL
FFF               000         000 RRR    RRR      RRR    RRR            TTT        LLL
FFF               000         000 RRR        RRR  RRR        RRR        TTT        LLL
FFF               000         000 RRR        RRR  RRR        RRR        TTT        LLL
FFF               000         000 RRR        RRR  RRR        RRR        TTT        LLL
FFF                 000000000     RRR            RRR RRR            RRR   TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR            RRR RRR            RRR   TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR            RRR RRR            RRR   TTT        LLLLLLLLLLLLLLL
```

FILEID**FORRANDOM

```
FFFFFFFFFF    000000    RRRRRRRR    RRRRRRRR      AAAAAA    NN      NN  DDDDDDDD      000000    MM      MM
FFFFFFFFFF    000000    RRRRRRRR    RRRRRRRR      AAAAAA    NN      NN  DDDDDDDD      000000    MM      MM
FF          00      00  RR      RR  RR      RR   AA    AA   NN      NN  DD      DD  00      00  MMMM  MMMM
FF          00      00  RR      RR  RR      RR   AA    AA   NN      NN  DD      DD  00      00  MMMM  MMMM
FF          00      00  RR      RR  RR      RR   AA    AA   NNNN    NN  DD      DD  00      00  MM  MM  MM
FF          00      00  RR      RR  RR      RR   AA    AA   NNNN    NN  DD      DD  00      00  MM  MM  MM
FFFFFFFF    00      00  RRRRRRRR    RRRRRRRR     AA    AA   NN  NN  NN  DD      DD  00      00  MM      MM
FFFFFFFF    00      00  RRRRRRRR    RRRRRRRR     AA    AA   NN  NN  NN  DD      DD  00      00  MM      MM
FF          00      00  RR  RR      RR  RR       AAAAAAAAAA   NN    NNNN  DD      DD  00      00  MM      MM
FF          00      00  RR  RR      RR  RR       AAAAAAAAAA   NN    NNNN  DD      DD  00      00  MM      MM
FF          00      00  RR    RR    RR    RR     AA    AA   NN      NN  DD      DD  00      00  MM      MM      ....
FF          00      00  RR    RR    RR    RR     AA    AA   NN      NN  DD      DD  00      00  MM      MM      ....
FF            000000    RR      RR  RR      RR   AA    AA   NN      NN  DDDDDDDD      000000    MM      MM      ....
FF            000000    RR      RR  RR      RR   AA    AA   NN      NN  DDDDDDDD      000000    MM      MM      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  FOR$RANDOM      ; random number generator and interfaces
0000     2          .IDENT  /1-003/         ; File: FORRANDOM.MAR  Edit: SBL1003
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28 ;
0000    29 ; FACILITY: FORTRAN SYSTEM LIBRARY
0000    30 ;++
0000    31 ; ABSTRACT:
0000    32 ;
0000    33 ;       Provide entry points for:
0000    34 ;               FOR$IRAN
0000    35 ;               FOR$RANDU_W
0000    36 ;
0000    37 ;       The algorithm used is copied exactly from PDP-11 FORTRAN
0000    38 ;       library so the same sequences will be generated.
0000    39 ;--
0000    40 ;
0000    41 ; VERSION: 1-001
0000    42 ;
0000    43 ; HISTORY:
0000    44 ;
0000    45 ; AUTHOR:
0000    46 ;       Jonathan M. Taylor, 12-Aug-77: Version 0
0000    47 ;
0000    48 ; MODIFIED BY:
0000    49 ;
0000    50 ;
0000    51 ;
```

```
0000     53          .SBTTL  HISTORY         ; Detailed Current Edit History
0000     54 ; Edit History for Version 0 of FOR$RANDOM
0000     55 ;
0000     56 ; 0-3   - use word offset to call for$JRAN TNH 16-SEP-77
0000     57 ; 0-4   - add a bug from 11 routine to make compatible:
0000     58 ;         - now tests only second parameter for 0 (first call),
0000     59 ;         - instead of concatenated longword JMT 6-OCT-77
0000     60 ; 0-5   - JRAN is now passed only one longword arg.  JMT 9-Oct-77
0000     61 ; 0-6   - Copy back seed as 2 words or 1 long word.  TNH 14-Nov-77
0000     62 ; 0-9   - Remove FOR$JRAN which is no longer supported.
0000     63 ;         FORTRAN compiler now generates calls to MTH$RANDOM.  JMT 4-Jan-78
0000     64 ; 0-10  - Bug fix 0-4 didn't break my code enough to be
0000     65 ;         compatable with the 11.  JMT 16-Feb-78
0000     66 ; 0-11  - Remove FOR$FLAG_JACKET.  TNH 11-July-78
0000     67 ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000     68 ; 1-002 - Add '' '' to the PSECT directive.  JBS 22-DEC-78
0000     69 ; 1-003 - Use .ENTRY.  SBL 1-Jul-1983
```

```
                     0000     71          .SBTTL  DECLARATIONS
                     0000     72
                     0000     73 ;
                     0000     74 ; INCLUDE FILES:
                     0000     75 ;       oerr.mar
                     0000     76
                     0000     77 ;
                     0000     78 ; EXTERNAL SYMBOLS:
                     0000     79 ;
                     0000     80
                     0000     81 ;
                     0000     82 ; MACROS:
                     0000     83 ;
                     0000     84
                     0000     85 ;
                     0000     86 ; PSECT DECLARATIONS:
                 00000000     87          .PSECT  _FOR$CODE       PIC, SHR, EXE, LONG, NOWRT
                     0000     88
                     0000     89 ;
                     0000     90 ; EQUATED SYMBOLS:
 00000004            0000     91          a1      =       4       ; offset into AP of address of arg1
 00000008            0000     92          a2      =       8       ; offset into AP of address of arg2
 0000000C            0000     93          a3      =       12      ; (optional) offset into AP of add-
                     0000     94                                  ; ress of output
                     0000     95
                     0000     96 ;
                     0000     97 ; OWN STORAGE:
                     0000     98 ;       NONE
```

```
                     100          .SBTTL  FOR$RANDU and FOR$RANDU_W       return number as parameter
                     101
                     102 ;++
                     103 ; FUNCTIONAL DESCRIPTION:
                     104 ;
                     105 ;       CALLs FOR$IRAN to get a random number and returns it in
                     106 ;       third parameter.
                     107 ;
                     108 ; CALLING SEQUENCE:
                     109 ;       CALL FOR$RANDU   (gen_base_1.ml.r, gen_base_2.ml.r,
                     110 ;                         random_fraction.wf.r)
                     111 ;
                     112 ;       CALL FOR$RANDU_W (gen_base_1.mw.r, gen_base_2.mw.r,
                     113 ;                         random_fraction.wf.r)
                     114 ;
                     115 ; INPUT PARAMETERS:
                     116 ;       gen_base_1              seed1 for algorithm
                     117 ;       gen_base_2              seed2 for algorithm
                     118 ;
                     119 ; IMPLICIT INPUTS:
                     120 ;       NONE
                     121 ;
                     122 ; OUTPUT PARAMETERS:
                     123 ;       random_fraction         floating point result is
                     124 ;                               between 0 and 1
                     125 ;
                     126 ; IMPLICIT OUTPUTS:
                     127 ;       NONE
                     128 ;
                     129 ; COMPLETION CODES:
                     130 ;       NONE
                     131 ;
                     132 ; SIDE EFFECTS:
                     133 ;       NONE
                     134 ;
                     135 ;--
                     136
                     137
                     138
                     139 FOR$RANDU_W::
                0000 0000 140          .ENTRY  FOR$RANDU, ^M<>
000C'CF    6C    FA 0002 141          CALLG   (AP), W^FOR$IRAN                ; R0 = floating result
  0C BC    50    50 0007 142          MOVF    R0, @a3(AP)             ; return as third parameter
                  04 000B 143          RET
```

```
                                        000C   145          .SBTTL  FOR$IRAN          result in R0
                                        000C   146
                                        000C   147 ;++
                                        000C   148 ; FUNCTIONAL DESCRIPTION:
                                        000C   149 ;
                                        000C   150 ;       SEED = arg1,arg2
                                        000C   151 ;       if arg2 = 0 then SEED = 1       ; first call only
                                        000C   152 ;       SEED = SEED * (2**16 + 3)
                                        000C   153 ;       arg1,arg2 = SEED                ; return for later calls
                                        000C   154 ;       R0 = SEED normalized to floating point
                                        000C   155 ;
                                        000C   156 ; CALLING SEQUENCE:
                                        000C   157 ;       Random_fraction.wf.v = FOR$IRAN (gen_base_1.mw.r,
                                        000C   158 ;                                       gen_base_2.mw.r)
                                        000C   159 ;
                                        000C   160 ; INPUT PARAMETERS:
                                        000C   161 ;       gen_base_1              seed1 for algorithm
                                        000C   162 ;       gen_base_2              seed2 for algorithm
                                        000C   163 ;
                                        000C   164 ; IMPLICIT INPUTS:
                                        000C   165 ;       NONE
                                        000C   166 ;
                                        000C   167 ; OUTPUT PARAMETERS:
                                        000C   168 ;       NONE
                                        000C   169 ;
                                        000C   170 ; IMPLICIT OUTPUTS:
                                        000C   171 ;       NONE
                                        000C   172 ;
                                        000C   173 ; COMPLETION CODES:
                                        000C   174 ;       NONE
                                        000C   175 ;
                                        000C   176 ; SIDE EFFECTS:
                                        000C   177 ;       NONE
                                        000C   178 ;
                                        000C   179 ; FUNCTIONAL VALUE:
                                        000C   180 ;       A floating-point value between 0 and 1
                                        000C   181 ;--
                                        000C   182
                                        000C   183
                                        000C   184
                                0000    000C   185          .ENTRY  FOR$IRAN, ^M<>
                  50    04 BC    B0    000E   186          MOVW    @a1(AP), R0             ; R0 = arg1
                50    50    10    9C    0012   187          ROTL    #16, R0, R0             ; build a longword value
                  50    08 BC    B0    0016   188          MOVW    @a2(AP), R0             ; in R0
                                        001A   189                                          ; NOTE: PDP-11 algorithm only checks
                                        001A   190                                          ; bits 15:0 for 0, so VAX is compatibile
                              23    13    001A   191          BEQL    20$                     ; branch if first call
51    50    00010003 8F    C5    001C   192          MULL3   #^X10003, R0, R1        ; R1 = R0 *((2**16)+3) = SEED
                  00 51    1F    E5    0024   193          BBCC    #31, R1, 15$            ; make sure SEED positive
                     50    51    4E    0028   194 15$::    CVTLF   R1, R0                  ; R0 = floating (SEED) binary point
                                        002B   195                                          ; to right of bit 0
      50    00003100 8F    44    002B   196          MULF    #^X3100, R0             ; R0 = R0 * 2**-31 = normalized, binary poin
                                        0032   197                                          ; to right of bit 31
                                        0032   198                                          ; R0 = floating point result
                                        0032   199                                          ; R1 = new seed
                  08 BC    51    B0    0032   200          MOVW    R1, @a2(AP)             ; return bits 15:0 of seed
                51    51    10    9C    0036   201          ROTL    #16, R1, R1             ; seed<31:16> to R1<15:0>
```

```
                04 BC    51    B0  003A   202          MOVW    R1, @a1(AP)             ; return bits 31:16 of seed as first arg
                                04  003E   203          RET                             ; return with R0 = floating random number
           50   00010000 BF    C0  003F   204 20$:     ADDL    #^X10000, R0            ; this is what the 11 did!
                   50    03    B0  0046   205          MOVW    #3, R0
                   51    50    D0  0049   206          MOVL    R0, R1
                         DA    11  004C   207          BRB     15$
                                   004E   208
                                   004E   209
                                   004E   210          .END
```

```
A1              = 00000004
A2              = 00000008
A3              = 0000000C
FOR$IRAN          0000000C RG    01
FOR$RANDU         00000000 RG    01
FOR$RANDU_W       00000000 RG    01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_FOR$CODE	0000004E (78.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.35
Command processing	120	00:00:00.50	00:00:02.42
Pass 1	68	00:00:00.59	00:00:01.29
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	50	00:00:00.44	00:00:01.94
Symbol table output	2	00:00:00.01	00:00:00.01
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	274	00:00:01.64	00:00:06.03

The working set limit was 900 pages.
2400 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 6 non-local and 2 local symbols.
210 source lines were read in Pass 1, producing 14 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

```
MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:FORRANDOM/OBJ=OBJ$:FORRANDOM MSRC$:FORRANDOM/UPDATE=(ENH$:FORRANDOM)
```

FORRAB
LIS

FORREADDF
LIS

FOROPNKEY
LIS

FORPAUSE
LIS

FORRANDOM
LIS

FOROPEN
LIS

FOROPENDE
LIS

FORREADDO
LIS